

Attorneys' Eyes Only

UNITED STATES DISTRICT COURT
NORTHERN DISTRICT OF CALIFORNIA
SAN FRANCISCO DIVISION

ORACLE AMERICA, INC.,)
Plaintiff,)
vs.) No. CV 10-03561 WHA
GOOGLE, INC.,)
Defendant.)

ATTORNEYS' EYES ONLY

Videotaped Deposition of EREZ LANDAU,
taken at 42 Chauncy Street, Boston,
Massachusetts, commencing at 9:50 a.m.,
Wednesday, September 14, 2011, before
Jill Shepherd, RPR, MA-CSR No. 148608,
NH-CSR No. 128, CA-CSR No. 13275, CLR,
Notary Public.

PAGES 1 - 113

Page 1

Veritext National Deposition & Litigation Services
866 299-5127

EXHIBIT 2-10

Attorneys' Eyes Only

<p>1 tape two. We are back on the record. 2 Q. To continue where we left off before the 3 break, let's take a look at paragraph 27 on 4 page six. You indicate that you modified 5 two files for your testing. One is 6 RuntimeInit.java and the second one is 7 AndroidRuntime.cpp; is that right? 8 A. Yes. 9 Q. Is RuntimeInit that Java part of the accused 10 functionality? 11 MR. PETERS: Objection. Form. 12 A. I want to open now, which is the accused 13 functionality basically is the fork, 14 disabling the fork -- disabling the usage of 15 fork as a standalone operation and move it 16 into fork/exec combination. When you're 17 using a new process in fork/exec folk, you 18 get disabling of the copy-on-write 19 functionality. So just looking right now 20 which one of the files was that. 21 One file contains -- both of them I 22 use. One file is calling the other 23 function, which is the defining the first 24 one. 25 Let me describe it for you. On</p> <p style="text-align: right;">Page 46</p>	<p>1 one place, and it was sufficient. 2 Q. Do you know which files are part of the 3 accused functionality? 4 MR. PETERS: Objection. Form. 5 A. I did not study it. 6 Q. Do you know if RuntimeInit.java is part of 7 the accused functionality? 8 MR. PETERS: Objection. Form. 9 A. As I was saying, I did not study it. My 10 task was only to disable it. 11 Q. You modified RuntimeInit.java, correct? 12 A. Yes. 13 Q. Do you know if that file was part of the 14 accused functionality before you modified 15 it? 16 MR. PETERS: Objection. Form. 17 A. You asked me this question -- this is the 18 third time you ask me. I say I don't know. 19 Q. Okay. 20 Did your modifications disable the 21 whole preloader? 22 MR. PETERS: Objection. Form. 23 A. What do you mean by "preloader"? In which 24 context? 25 Q. When the zygote starts up, it preloads a</p> <p style="text-align: right;">Page 48</p>
<p>1 page one of Appendix A, we added a new JNI 2 function, which is in the RuntimeInit exec, 3 which do the exec operation. The exec 4 operation disable the fork as a standalone, 5 and, by this, disable copy-on-write. So 6 together they are doing this operation. The 7 actual disabling is inside B and A is 8 calling B. 9 Q. A is RuntimeInit.java? 10 A. Yes. 11 Q. And B is AndroidRuntime.cpp? 12 A. Yes. No, B is AndroidRuntime.cpp. This 13 is A and this is B [indicating]. 14 Q. I'm sorry. Right. 15 Are any other files part of the 16 accused functionality? 17 MR. PETERS: Objection. Form. 18 A. Can you -- are there any other files what? 19 Q. That are part of the accused functionality 20 in Android. 21 MR. PETERS: Objection. Form. 22 A. I was not looking for the accused 23 functionality. I was looking how to disable 24 it. It's the actual functionality is 25 composed many places, but I disable only in</p> <p style="text-align: right;">Page 47</p>	<p>1 number of frameworks into memory; is that 2 correct? 3 MR. PETERS: Objection. Form. 4 A. You also do that. This is not the only 5 thing it does. 6 Q. It is one of the functions that it performs; 7 is that correct? 8 MR. PETERS: Objection. Form. 9 A. Correct. 10 Q. And those frameworks that are loaded can be 11 cloned using copy-on-write in Android; is 12 that your understanding? 13 MR. PETERS: Objection. Form. 14 A. Your description of the situation is not 15 precise. 16 Q. Let me start over then. 17 Does Android have a preloader? 18 MR. PETERS: Objection. Form. 19 A. Preloader is used in several places in 20 different contexts. This is why I ask you 21 what context you are referring. 22 Q. Do Java virtual machines have a class 23 preloader? 24 MR. PETERS: Objection. Form. 25 A. Again, same answer. Preloader is not a</p> <p style="text-align: right;">Page 49</p>

13 (Pages 46 - 49)

Attorneys' Eyes Only

<p>1 come to Israel.</p> <p>2 MR. PETERS: I have a few follow-up</p> <p>3 questions, Mr. Landau.</p> <p>4 * * * * *</p> <p>5 EXAMINATION BY MR. PETERS</p> <p>6 Q. If you look at Exhibit 496, which is a</p> <p>7 document entitled "MtaskClassLoading."</p> <p>8 A. The list of class loading? Yes. Found it.</p> <p>9 Q. This is your program that will load in the</p> <p>10 classes that are named here in the static</p> <p>11 stream classes list on the first page?</p> <p>12 A. Yes.</p> <p>13 Q. If I could ask you to turn to page 21,</p> <p>14 please.</p> <p>15 A. Yes.</p> <p>16 Q. And here at the bottom of page 21, there's a</p> <p>17 series of classes that are in the package</p> <p>18 space</p> <p>19 org.apache.harmony.xnet.provider.jsse.</p> <p>20 Do you see those?</p> <p>21 A. Yes.</p> <p>22 Q. So when your test program loads, it will</p> <p>23 load those classes; is that correct?</p> <p>24 A. It can only load them. It cannot execute</p> <p>25 them.</p> <p style="text-align: right;">Page 106</p>	<p>1 Q. Is there any relationship between your test</p> <p>2 program loading that class and the</p> <p>3 bluetooth.ScoSocket entries -- in the log</p> <p>4 entry of Exhibit 499?</p> <p>5 A. In the Bluetooth case it's likely that you</p> <p>6 are right, but we have to verify this in the</p> <p>7 code.</p> <p>8 Q. So it could be not accidental that these log</p> <p>9 entries relating to bluetooth.ScoSocket</p> <p>10 appear in the log because of the loading of</p> <p>11 the class ScoSocket?</p> <p>12 A. It could be. We have to check it.</p> <p>13 Q. Speaking more generally about the tests that</p> <p>14 you and your team ran, would you say that</p> <p>15 the tests that are described in your report</p> <p>16 and you've testified about today, are they</p> <p>17 representative of the kind of performance</p> <p>18 testing that you and your team members do in</p> <p>19 your everyday work?</p> <p>20 A. Yes. This is a typical scenario of work</p> <p>21 that we are doing.</p> <p>22 Q. Okay.</p> <p>23 And when you do your work, do you</p> <p>24 normally rely on your colleagues like Seon</p> <p>25 to help you do your work?</p> <p style="text-align: right;">Page 108</p>
<p>1 Q. But what will it do with any sort of static</p> <p>2 initialization methods that may be present</p> <p>3 when it initializes the classes, when the</p> <p>4 classes are loaded?</p> <p>5 A. In the 4.8, I don't think it's been done.</p> <p>6 I'm not sure about this. Anyway, this is</p> <p>7 not a message you get over there -- from the</p> <p>8 log. The message in the log is not due to</p> <p>9 that.</p> <p>10 Q. So there's -- so in your view, there's no</p> <p>11 relationship between the loading of the</p> <p>12 classes and the log message in Exhibit 499?</p> <p>13 A. It could be. I would say it's not. It</p> <p>14 could be.</p> <p>15 Q. Okay.</p> <p>16 If I could ask you to turn to page two</p> <p>17 of Exhibit 496, MtaskClassLoading.java.</p> <p>18 A. Which one? The same one?</p> <p>19 Q. Yes. Page two.</p> <p>20 So here do you see a series of classes</p> <p>21 all beginning the name "android.bluetooth"?</p> <p>22 A. Yes.</p> <p>23 Q. And the last one on that list is</p> <p>24 "android.bluetooth.ScoSocket"?</p> <p>25 A. Yes.</p> <p style="text-align: right;">Page 107</p>	<p>1 A. I assist a lot of and -- I am being assisted</p> <p>2 a lot with hands-on work with other of</p> <p>3 the -- I'm supervising quite multiple</p> <p>4 projects.</p> <p>5 Q. Is there anything about the projects that</p> <p>6 you did in your analysis of Android that you</p> <p>7 would say is not typical of the kinds of</p> <p>8 testing you do in the Java environment?</p> <p>9 A. No. We use normal methods that -- normal</p> <p>10 procedure.</p> <p>11 Q. Okay.</p> <p>12 MR. PETERS: Thank you, Mr. Landau.</p> <p>13 THE VIDEOGRAPHER: The time is</p> <p>14 1:34 p.m. This is the end of tape three,</p> <p>15 and the deposition is concluded, and we are</p> <p>16 off the record.</p> <p>17 (Whereby the proceedings were</p> <p>18 concluded at 1:34 p.m.)</p> <p>19</p> <p>20</p> <p>21</p> <p>22</p> <p>23</p> <p>24</p> <p>25</p> <p style="text-align: right;">Page 109</p>

28 (Pages 106 - 109)